T. Fondrat

PROGRAMMING WITH WORD

write Visual Basic programs using MS Word

Not many PC users are aware that the well-known Microsoft Word includes a programming environment for Visual Basic. It allows you to get to grips with the fundamentals of this programming language.

This article will be of particular interest to those of you who don't have the complete Visual Basic package, but who would still like to play with a programming language under Windows. The trick here is: use **Visual Basic for Applications**, which is available in Word as well as Excel.

Step 1

Start Word. All the screendumps and programs shown here were made using Word 2000, but other versions such as Word 97 work just as well!

Step 2

From the *View* menu click on *Toolbars*. Then select *Visual Basic*, as shown in figure 1.

Step 3

The *Visual Basic* toolbar appears (figures 2 and 3).

Step 4

Now we'll make a button called **CommandButton** (figure 4). To look at a list of the properties of the button you have to position the mouse cursor over the button, then right-click on it and select *Properties*.

A new window pops up containing all the properties of the object (figure 5).
The two most important properties are *name* and *caption*.

Give a name to the button, which in our example is **ButtonTestMessage**.
Replace the text on the button with **Click on this button!**.
You can also change other settings, such as the size, font, background colour, background picture, etc.

As an aside, Visual Basic has many windows that are clearly laid out, or not! That is left to the user, but try not to get confused by all those open windows.

Step 5

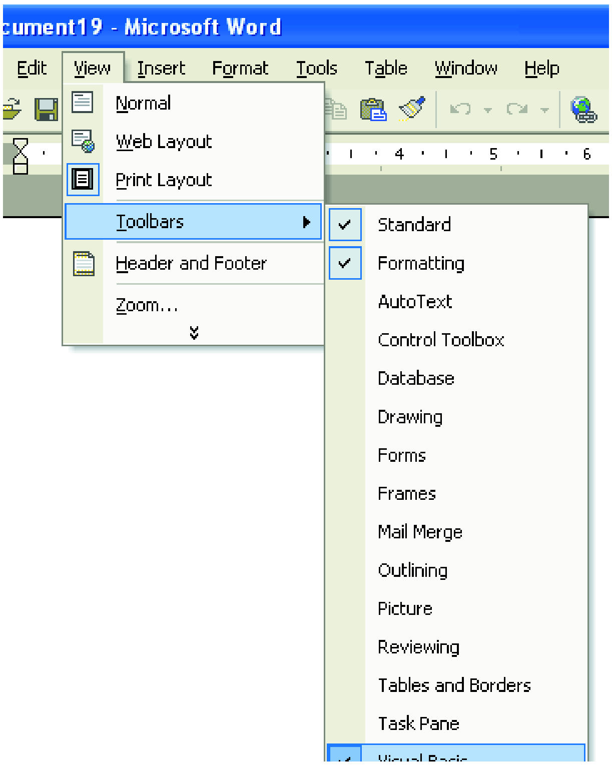
Double-click on the button.
The Visual Basic editor opens. The procedure *ButtonTestMessage_Click* is created, if it didn't exist already, and the cursor appears HERE, where you will write your program (figure 6)!

This procedure is run with every click on the **ButtonTestMessage** button.

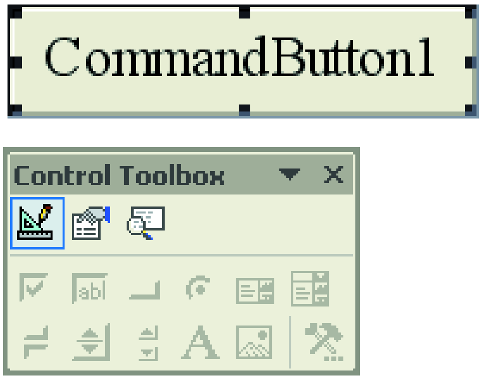
Or more precisely, the procedure *Click* is linked to the object **ButtonTestMessage**, of type *CommandButton* (also described as: belongs to the CommandButton class).
The code for this procedure is entered here.

Step 6

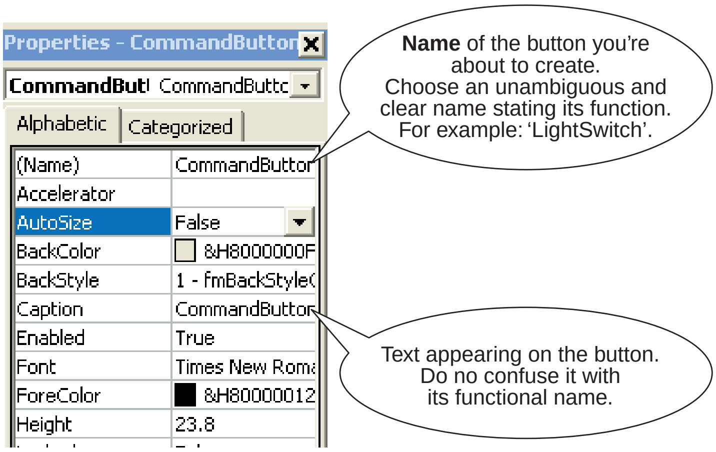
Enter the line: *MsgBox ("You've clicked here!")*
(figure 7a).



1



4

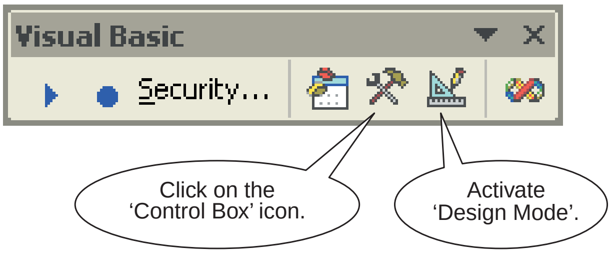


(Name)	CommandButtor
Accelerator	
AutoSize	False
BackColor	&H8000000F
BackStyle	1 - fmBackStyle(
Caption	CommandButtor
Enabled	True
Font	Times New Roma
ForeColor	&H80000012
Height	23.8

Name of the button you're about to create. Choose an unambiguous and clear name stating its function. For example: 'LightSwitch'.

Text appearing on the button. Do no confuse it with its functional name.

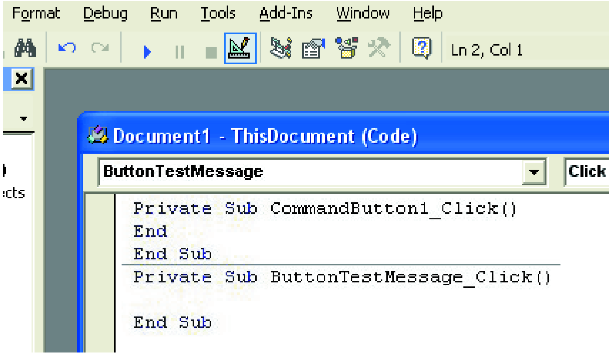
5



Click on the 'Control Box' icon.

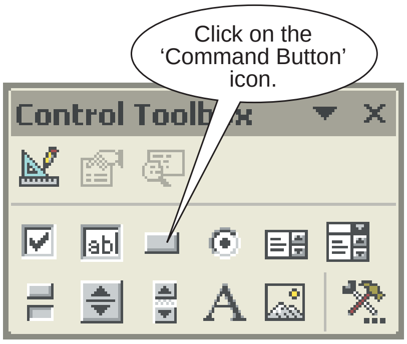
Activate 'Design Mode'.

2



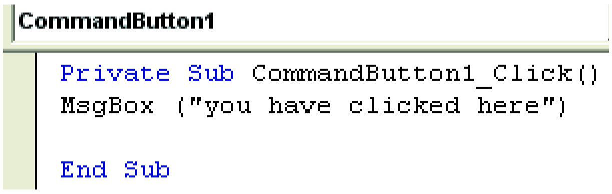
```
Private Sub CommandButton1_Click()
End
End Sub
Private Sub ButtonTestMessage_Click()

End Sub
```

6



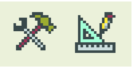
Click on the 'Command Button' icon.

3



```
Private Sub CommandButton1_Click()
MsgBox ("you have clicked here")

End Sub
```

7a



7b

Step 7
Return to Word, using Alt-Tab for example, and return to Run-mode by deselecting the two icons on the Visual Basic toolbar (figure 7b).

You can also minimise the *Visual Basic* toolbar if you want to have a clear desktop, but you will have to enable it again if you want to continue programming.

Step 8
Save the file in Word format and test how it runs.

Summary

A small program can be written in seconds using MS Word. You should now have an idea of all the possibilities it offers. The same method can also be used in MS Excel to program in Visual Basic: create windows with buttons, input fields for text or data, pictures, etc. Show the information in graphs, personalise the Word menu with user-defined commands, run other programs. It is also possible (although that really needs a separate article) to use API's under Windows, to program the parallel port for example. Complex programs that hide the standard menus can be written this way. The users will think that they're running a dedicated program, whereas the reality is that it runs under Word or Excel.

(040046-1)

Help, my program doesn't work!

Three possible solutions and some advice.

1.Security level.
If a button doesn't work you should first check that the security level has been set to 'Medium'. The setting of this level determines how Word deals with macros and Visual Basic code that are embedded in documents. Word always asks this question when a document is opened. A high security level gives you no choice and programs written in Visual Basic cannot be opened.

The security level in Word can be changed as follows:

– From the Tools menu click on **Macro** and then on **Security**.

– Select the **Medium** *security level.*

– Now save your work and close Word.

– Re-open the document.

When the document is opened, Word gives a warning that it contains macros. In this instance, we click on **Enable Macros**.

2. Problems when running a program.
Your program could contain a bug, which has to be tracked down.

The debugger is a utility that stops your program at the point where the error occurs! The screendump below shows a typical window that you could see.

The Visual Basic editor then opens automatically.

3. It still doesn't work.
It's possible that the cause of the problem lies elsewhere. Try the following: Save the program and close Word. Open the document again. Amazingly, this trick is often enough to get a program to work.

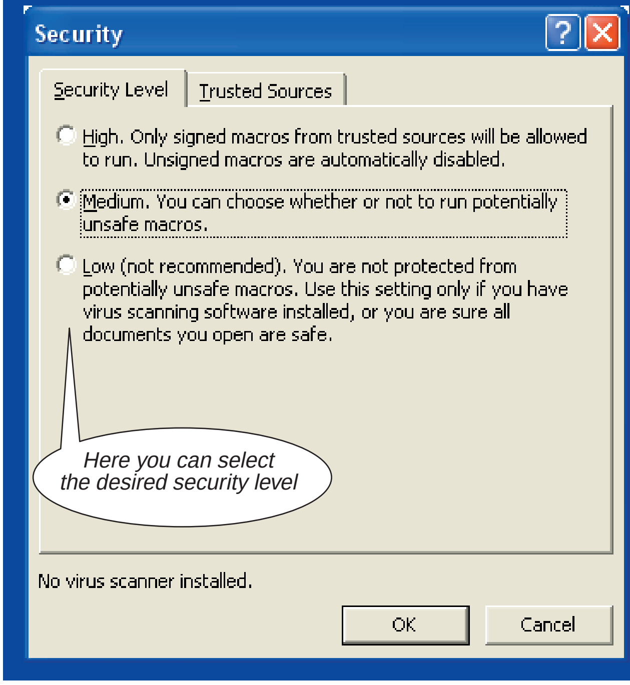
If none of these help, there is a large amount of information on various Internet sites and in discussion forums.

Internet links:

Further information on Visual Basic
http://msdn.microsoft.com/vbasic/

Visual Studio
http://msdn.microsoft.com/vstudio/

Source codes in VB
http://www.vbcode.com/



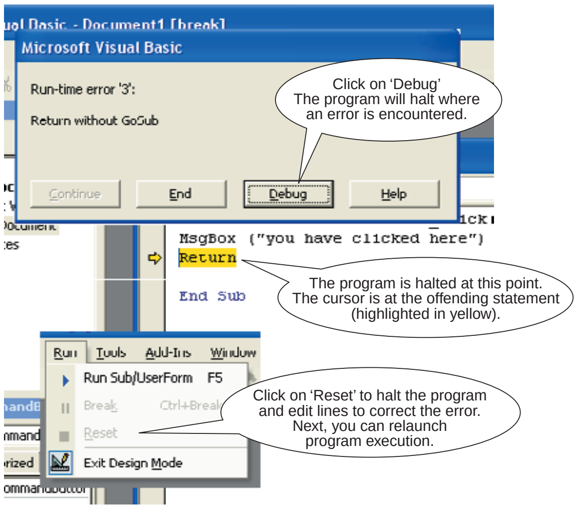
○ High. Only signed macros from trusted sources will be allowed to run. Unsigned macros are automatically disabled.

● Medium. You can choose whether or not to run potentially unsafe macros.

○ Low (not recommended). You are not protected from potentially unsafe macros. Use this setting only if you have virus scanning software installed, or you are sure all documents you open are safe.

No virus scanner installed.

Here you can select the desired security level



Run-time error '3':

Return without GoSub

```
MsgBox ("you have clicked here")
Return

End Sub
```

Click on 'Debug'
The program will halt where an error is encountered.

The program is halted at this point.
The cursor is at the offending statement (highlighted in yellow).

Click on 'Reset' to halt the program and edit lines to correct the error.
Next, you can relaunch program execution.